NAWCWPNS TP 8217
Revision 1

Multilayer Piecewise Linear Networks

by
W. O. Alltop
Research Department

NOVEMBER 1994

NAVAL AIR WARFARE CENTER WEAPONS DIVISION
CHINA LAKE, CA 93555-6001

Naval Air Warfare Center Weapons Division

FOREWORD

This report presents techniques for weight initialization in piecewise linear neural networks. The work was performed during 1993 and 1994 as part of the Office of Naval Research Independent Research Program.

This paper is being revised because of a transformation of characters on the computer.

Approved by
R. L. DERR, *Head*
Research and Technology Division
21 November 1994

Under authority of
D. B. McKinney
RAdm., U.S. Navy
Commander

Released for publication by
S. HAALAND, *Head,*
NAWCWPNS Engineering Group

NAWCWPNS Technical Publication 8217 Revision 1

Published by . . . . Scientific and Technical Documentation
Collation . . . . Cover, 28 leaves
First printing . . . . 40 copies

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	July 1994	FINAL- 1993-1994

4. TITLE AND SUBTITLE
MULTILAYER PIECEWISE LINEAR NETWORKS

5. FUNDING NUMBERS
N00014-94-WX-41012
61152N

6. AUTHOR(S)
W. O. Alltop

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Air Warfare Center Weapons Division
China Lake, CA 93555-6001

8. PERFORMING ORGANIZATION REPORT NUMBER
NAWCWPNS TP 8217 Revision 1

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
800 N. Quincy
Arlington, VA 22217

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12A. DISTRIBUTION/AVAILABILITY STATEMENT
A Statement; public release; distribution unlimited

12B. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

(U) Feed-forward layered networks are the simplest neural computers. These networks provide fast compact devices for computing discriminant functions and other mappings applicable to pattern recognition problems. This report treats the problem of weight initialization/assignment for feed-forward networks with piecewise linear neuron transfer function. Specific transformations are shown to be impossible with a single hidden layer. Techniques are presented for constructing such transformations with two hidden layers.

14. SUBJECT TERMS
Neural Networks
Pattern Recognition
Geometry

15. NUMBER OF PAGES
54

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

CONTENTS

1. INTRODUCTION

The piecewise linear layered neural network is a simple computing device with potential for implementing pattern recognition and image processing algorithms. Questions regarding mapping capabilities and weight assignment for these networks lead to problems in combinatorial and computational geometry, due to the discrete and essentially linear nature of the piecewise linear neuron transfer function.

This report discusses some specific techniques and results for piecewise linear networks (PLNs). The basic problems motivating this discussion concern the need to design networks and assign their weights in order to obtain network transformations which map specified input vectors into specified output vectors. For example one may have a sample prototype in d-dimensional space for each of N pattern classes. The objective may then be to map the ith prototype x_i into a specified m-dimensional vector y_i, for $1 \leq i \leq N$.

Given d, m, N, and the N pairs (x_i, y_i), how does one determine the number of hidden layers and their dimensions for a suitable layered network? This is the network design problem. Given a network of specified type, how does one then determine a set of weights that will map x_i into y_i for all i's? The second problem regards weight assignment. Recent work in weight assignment has focused largely on iterative algorithms like back propagation. Our focus is on methods for weight initialization and assignment which avoid costly iterative procedures.

References 1 and 2 establish relationships between the dimensions of the network layers and the numbers of pairs which can be accommodated. References 2 through 4 give examples of noniterative weight assignment procedures. The methods of References 2 and 3 employ linear algebraic techniques which are effective for large classes of well-behaved sigmoidal neuron transfer functions. The method of Reference 4 utilizes convexity properties as well as affine geometry, and applies only to the piecewise linear neuron transfer function.

Basic results from linear algebra and convexity can be found in References 5 and 6. The fundamentals of combinatorial and computational geometry are presented in References 7 and 8. Separation and mapping capabilities of layered networks are discussed in References 9 through 12. Reference 13 contains the fundamental material on multidimensional order types.

Section 2 presents basic definitions and notation. Several concepts from geometric complexity are discussed in Section 3. These include the interior relation (INT), dichotomies and decomposition by hyperplanes. A construction for (d,2,m) mappings is also given. Section 4 contains two theorems pertaining to order modification by PLNs, as well as two examples of (2,2,2,2) PLN mappings on sets of five planar points.

2. DEFINITIONS AND NOTATION

All patterns reside in real affine spaces. The layer-to-layer mappings are compositions of affine transformations and the coordinate-wise neuron transfer function. Unless stated otherwise, we assume throughout that the neuron transfer (squashing) function is the piecewise linear function p, defined by

$$p(t) = \left.\begin{array}{l} -1 \; for \; t < -1 \\ t \; for \; -1 \le t < 1 \\ 1 \; for \; 1 \le t \end{array}\right\} .$$

The function p is extended to vectors in a coordinate-wise fashion. That is,

$$p(x) = \left(p(x_1), p(x_2), \ldots, p(x_d)\right)$$

where

$$x = \left(x_1, x_2, \ldots, x_d\right) .$$

$R^{(d)}$ denotes d-dimensional real affine space while $I^{(d)}$ denotes the d-dimensional real cube $[-1,1]^{(d)}$. The input set X and the desired output set Y are assumed to be in general position in $R^{(d)}$ and $I^{(d)}$, respectively. An $(L_0,L_1,\ldots,L_K,L_{K+1})$-network is a feed-forward layered network with

input dimension = d = L_0

output dimension = m = L_{K+1}

and

K hidden layers with dimensions = L_j, $1 \le j \le K$.

The nodes in layer j are forward connected to those in layer j+1 for $0 \le j \le K$. For economy of notation we let

$$L^* = (L_0, L_1, L_2, \ldots, L_K, L_{K+1}) \quad .$$

We say that L^* accommodates an integer N, if for every pair (X, Y) of N distinct inputs and N desired outputs there exists a weight assignment for an L^*-network which effects the mapping $x_i \rightarrow y_i$, for $1 \leq i \leq N$. Here the sets X and Y are assumed to be in general position in $R^{(d)}$ and $R^{(m)}$, respectively. $N_{max}(L^*)$ denotes the largest integer N which is accommodated by L^*.

Among the N-sets of input/output pairs are those whose output sets lie in the interior of the cube $I^{(d)}$. Any mapping that accommodates such a set makes no use of the piecewise linear truncation in the output space. Thus, N_{max} has the same value without the final 'squash' as with it. Therefore, we will usually omit the application of the function p at the output layer.

The mapping from layer j to j+1 is given by

$$A_j^+(z) = p(A_j z + b_j)$$

where A_j is L_{j+1} by L_j, and b is L_{j+1} by 1.

The total number of weights available in a network of type L^* is denoted by $Wgt(L^*)$ and is given by

$$Wgt(L^*) = (d+1)L_1 + (L_1+1)L_2 + \ldots + (L_{K-1}+1)L_K + (L_K+1)m$$

$N_{dim}(L^*)$ is an upper bound for $N_{max}(L^*)$, established in Reference 1, and given by

$$N_{\dim}(L^*) = Wgt(L^*)/m \quad .$$

A subset C of $R^{(d)}$ is called convex provided $\lambda_1 c_1 + \lambda_2 c_2$ C, whenever $c_1 \in$ C, $c_2 \in$ C, $\lambda_1 \geq 0$, $\lambda_2 \geq 0$, and $\lambda_1 + \lambda_2 = 1$. Equivalently C is convex if, and only if, C is closed under convex combinations. If C is topologically closed and convex, the boundary of C, denoted Bound(C), is the topological boundary of C in the topology of Aff(C). Aff(C) denotes the affine closure of C, i.e. the smallest affine subspace of $R^{(d)}$ containing C. The interior of C, denoted Int(C), is just C\Bound(C). A point c is an extreme point of C whenever there exists a hyperplane H in Aff(C) for which $H \cap C = \{c\}$, and H does not separate C. Note that if Aff(C) is k-dimensional, then a hyperplane H in Aff(C) must be a

(k-1)-dimensional affine subspace of $R^{(d)}$ which lies in Aff(C). The set of extreme points of C is denoted Ext(C).

EXAMPLE 1

Let d = 3, and let

$$C = \left\{(c_1, c_2, c_3): c_1^2 + c_2^2 \leq 1 \textit{ and } c_3 = 1\right\}$$

The topological boundary of C in $R^{(d)}$ is C. Since Aff(C) is the hyperplane $\{(x_1, x_2, x_3) : x_3 = 1\}$, we have

$$\text{Bound(C)} = \{(c_1, c_2, 1) : c_1^2 + c_2^2 = 1\},$$

$$\text{Int(C)} = \{(c_1, c_2, 1) : c_1^2 + c_2^2 < 1\} \quad ,$$

and

$$\text{Ext(C)} = \text{Bound (C)}.$$

Therefore, Bound(C) is the circle, Int(C) is the open disk, and the set of extreme points is also the circle.

The last equality does not generally hold. Closed convex polytopes in $R^{(d)}$ have (d-1)-dimensional boundaries, but only finite 0-dimensional sets of extreme points, which are called vertices.

EXAMPLE 2

Let d = 3, and let

$$C = \{(c_1, c_2, c_3) : \text{all } c_i \geq 0 \text{ and } c_1 + c_2 + c_3 = 1\} \ .$$

C is just the 2-simplex embedded in $R^{(3)}$. In this case

$$\text{Bound(C)} = \{(c_1, c_2, c_3) \in C : \text{some } c_i = 0\}.$$

$$\text{Int(C)} = \{(c_1, c_2, c_3) \in C : \text{all } c_i > 0\},$$

and

$$\mathrm{Ext}(C) = \{c_1, c_2, c_3) \in C : \text{some } c_i = 1\} \ .$$

In this example Bound(C) is the one-dimensional boundary of the triangle, while Ext(C) consists only of the three vertices.

For $X \subset R^{(d)}$, the convex closure of X, denoted Conv(X) or Hull(X), is the intersection of all convex sets which contain X. Let C(X) = $\mathrm{Conv}^{-}(X)$, the topological closure of Conv(X) in $R^{(d)}$. We define Bound(X), Int(X), and Ext(X) in terms of the closed convex set C(X):

$$\mathrm{Bound}(X) = X \cap \mathrm{Bound}(C(X)),$$

$$\mathrm{Int}(X) = X \cap \mathrm{Int}(C(X)),$$

and

$$\mathrm{Ext}(X) = X \cap \mathrm{Ext}(C(X)).$$

EXAMPLE 3

Let d = 2, and define X, a set of nine planar lattice points, by

$$X = \{(x_1, x_2) : -1 \le x_i \le 1 \text{ and } x_i \text{ an integer}, i = 1, 2\}$$

Then

$$\mathrm{Bound}(X) = \{(-1, -1), (-1, 0), (-1, 1), (0, -1), (0, 1), (1, -1), (1,0),(1, 1)\}$$

$$\mathrm{Int}(X) = \{(0, 0)\}$$

and

$$\mathrm{Ext}(X) = \{(-1, -1), (-1, 1), (1, -1), (1, 1)\} \ .$$

X has four vertices, four other boundary points, and one interior point.

In this report we will be primarily interested in finite sets X in general position in $R^{(d)}$. For such sets, Bound(X) = Ext(X) = the set of vertices of the polytope Conv(X) and Int(X) consists of all remaining points of X.

The following nest of subsets is associated with each subset $X \subset R^{(d)}$:

$$\mathrm{Ext}(X) \subseteq \mathrm{Bound}(X) \subseteq X \subseteq \mathrm{Conv}(X).$$

Suppose X and Y satisfy the following:

$$X = \{x_i : 1 \le i \le N\} \subset R^{(d)},$$

$$Y = \{y_i : 1 \le i \le N\} \subset R^{(e)}$$

where

$$y_i = A^+(x_i)\,,\ 1 \le i \le N\,,$$

and $A^+ : R^{(d)} \rightarrow R^{(e)}$ is an affine transformation. Then $x \in \mathrm{Bound}(X)$ whenever $y \in \mathrm{Bound}(Y)$, i.e.

$$\mathrm{Bound}(A^+(X)) \subseteq A^+\,(\mathrm{Bound}(X))$$

If A^+ is injective (1-1), then equality holds.

Table 1 contains the coefficients of two affine mappings A_1^+ and A_2^+ from $R^{(2)}$ to $R^{(2)}$. A_2^+ is bijective but A_1^+ is not. Table 2 gives the coordinates of three 4-sets X, Y, and Z in $R^{(2)}$, satisfying $Y = A_1^+(X)$, $Z = A_2^+(X)$. Figure 1 shows the three 4-sets in $R^{(2)}$. A_1^+ is not bijective, and the boundary point x_3 in X goes to an interior point y_3. The second mapping A_2^+ is bijective, and the boundary $\{x_1, x_2, x_3\}$ maps onto the boundary $\{z_1, z_2, z_3\}$.

TABLE 1. Two Affine Mappings.

$$A_1^+\left((s,t)^T\right) = \begin{bmatrix} 0.5s + t - 2 \\ s + 2t - 4 \end{bmatrix}$$

$$A_2^+\left((s,t)^T\right) = \begin{bmatrix} s - t - 2 \\ s + t - 1 \end{bmatrix}$$

TABLE 2. Three Planar 4-Sets.

i	x_i^T	y_i^T	z_i^T
1	(0, 6)	(4, 8)	(-8, 5)
2	(-4, 0)	(-4, -8)	(-6, -5)
3	(8, -1)	(1, 2)	(7, 6)
4	(2, 1)	(0, 0)	(-1, 2)

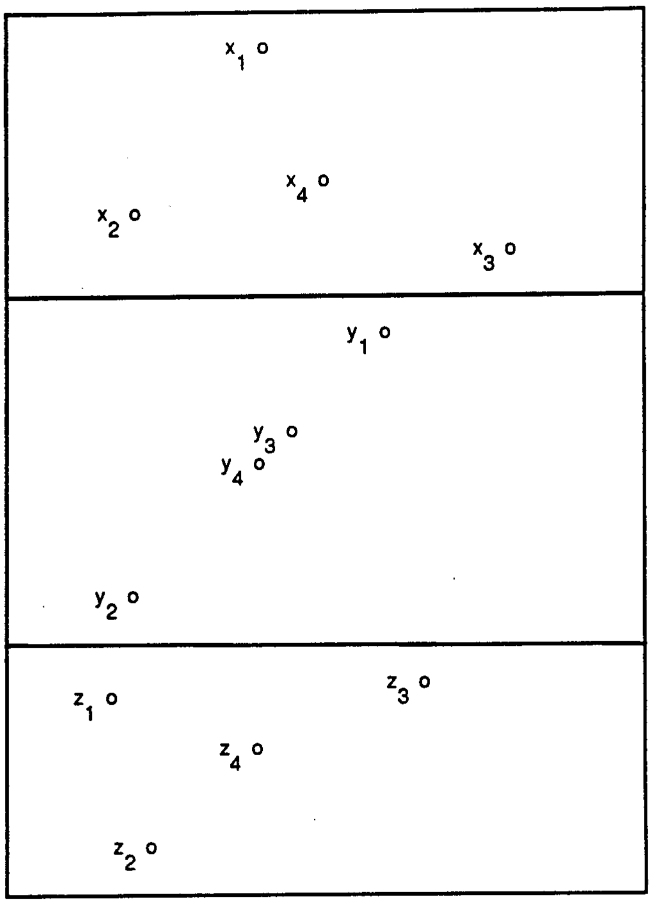

FIGURE 1. Three Planar 4-Sets.

When N > d > e, and X is in general position in $R^{(d)}$, then there always exists a linear mapping A, with maximal rank e, such that Bound (A(X)) ≠ A(Bound(X)). Indeed, for any $x_0 \in$ Ext(X), A can be selected so that $A(x_0)$ is an interior point of A(X). To prove this we select a point c in Conv(X) such that X ∪ {c} is in general position, and let

$h_0 = x_0 - c$. Since $h_0 \neq 0$, the subspace $H = h_0^{\perp}$, perpendicular to h_0, is a hyperspace in $R^{(d)}$. Therefore, $\dim(H) = d-1 \geq e$, so we may choose a set $\{h_1, h_2, ..., h_e\}$ of e linearly independent vectors in H. The desired transformation A is then defined by

$$A(x) = [h_1, h_2, \ldots, h_e]^T x$$

Clearly A maps $R^{(d)}$ to $R^{(e)}$ and has maximal rank e. Moreover, since $A(x_0) = A(c)$, $A(x_0)$ must be an interior point of A(X). Selected perturbations of c and the h_i's allow A to be defined so that A(X) is in general position in $R^{(e)}$.

3. PROJECTIONS

The piecewise linear function, although quite simple to define and visualize, delivers considerable complexity when composed with affine mappings and iterated. The space of affine mappings is closed under composition and preserves certain convexity properties of subsets of the domain. In particular, the relations

$$\text{(INT)} \qquad x_i \in \mathrm{Int}\,(\{x_1, x_2, ..., x_N\})$$

are preserved by injective linear mappings. The modification of these properties provides the basis for the increased capabilities of nonlinear networks.

Consider, for example, the (1,L,1)-network. If the neuron transfer function is linear, then the network transfer function preserves the (INT) relation. In $R^{(1)}$ this means that the function must be monotone. By contrast the (1,L,1)-PLN can produce up to L-1 local maxima and L-1 local minima. This imposes an upper bound of 2L on N_{max} (1,L,1). Surprisingly N_{max} (1,L,1) is actually equal to 2L (Reference 12). For more general squashing functions one can accommodate L+1 pairs using straight forward linear algebraic techniques (References 3 and 4). It is also known that 2L-1 pairs can be approximated using smooth sigmoid (Reference 12). Thus, for the general sigmoid there is quite a gap between the number of pairs that can be accommodated exactly and the number that can be approximated using known methods.

Comparing the (1,L,1)-PLN to the perceptron is perhaps more interesting. In perceptrons the neurons employ the threshold transfer function T, defined by

$$T(t) = \left.\begin{matrix} -1 \; for \; t < 0 \\ 1 \; for \; 0 \leq t \end{matrix}\right\} \; .$$

Realistic comparison requires criteria other than bounds for N_{max}. The outputs at each hidden layer of a perceptron all lie at the vertices of the cube. This greatly restricts the possible output sets for a perceptron. In particular, the set of outputs Y must all lie in the boundary of Conv(Y), i.e. Bound (Y) = Y. This means that no set of pairs can be accommodated by a perceptron when Int(Y) is non-empty. Thus, for the (d,L,m)-perceptron, $N_{max} \leq m + 1$.

Another measure of capability often applied to families of real-valued functions is derived from the notion of realizable dichotomies (References 9 and 12). A set of points is accommodated by a network provided all dichotomies are realizable. Alternatively such a set is said to be shattered by the network. A dichotomy of X is just a decomposition of X into two disjoint subsets X_1, X_2. The dichotomy $\{X_1, X_2\}$ is realizable by a family F of real-valued functions provided there exists $f \in F$ satisfying:

$$f(x_1) < 0 < f(x_2) \text{ whenever } x_1 \in X_1 \text{ and } x_2 \in X_2,$$

or

$$f(x_2) < 0 < f(x_1) \text{ whenever } x_1 \in X_1 \text{ and } x_2 \in X_2.$$

An integer N is now said to be accommodated by F provided every set of N points is shattered by F. The maximum value of N that can be accommodated, in this sense, by a family of functions is denoted by N_{dich}. In this setting, at most L+1 points can be accommodated in general by a (1,L,1)-perceptron. Suppose X is an (L+2)-subset of R. Let

$$X = \{x_1, x_2, ..., x_{L+2}\},$$

where

$$x_1 < x_2 < .... < x_{L+2}.$$

Now let $X_1 = \{x_{2k+1} : 1 \leq 2k+1 \leq L+2\}$, and $X_2 = \{x_{2k} : 2 \leq 2k \leq L+2\}$. The components X_1 and X_2 of the dichotomy are interleaved on the line. Every interval (x_i, x_{i+1}) must be cut by one of the hidden neurons. Since there are L+1 intervals and only L neurons, this is not possible.

Using realizable dichotomies for measuring network mapping capability yields

$$N_{dich} = \left.\begin{matrix} L+1 \; for \; (1,L,1) - perceptrons \\ 2L \; for \; (1,L,1) - PLNs \end{matrix}\right\}.$$

This comparison shows a factor of 2 increase in capability of the (1,L,1)-PLN over the (1,L,1)-perceptron. This type of comparison is treated in more detail in Reference 12.

The network function for a PLN is piecewise affine. That is, for any weight assignment W, there is a decomposition of the domain $R^{(d)}$ into convex sets, with disjoint interiors, on each of which the network function F_W is affine. Table 3 shows the weights for a (2,3,1)-PLN. The decomposition of the domain into 19 'affine pieces' is shown in Figure 2. Figure 3 shows the 21 pieces which result when the squashing function is also applied in the output space. For a (2,L,1)-network the number of distinct affine regions in $R^{(2)}$ can be as great as $2L^2 + 1$ without squashing in the output space.

TABLE 3. Thirteen Weights for a (2, 3, 1)-Network.

$$1st\ layer = \begin{bmatrix} 3 & -3 & 0 \\ 3 & 3 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$2nd\ layer = \begin{bmatrix} 1 & 1 & -2 & 0 \end{bmatrix}$$

$$(x_1, x_2) \to (u_1, u_2, u_3) \to y \to y'$$

$$u_1 = p(3x_1 - 3x_2)$$

$$u_2 = p(3x_1 + 3x_2)$$

$$u_3 = p(x_1)$$

$$y = u_1 + u_2 - 2u_3$$

$$y' = p(y)$$

Letting Aff(L^*) denote the number of affine regions possible in an L^*-network, without squashing in the output space, it can be shown that

$$Aff(d, L, 1) = \sum \left\{ 2^k \binom{L}{k} : 0 \le k \le d \right\}$$

Thus, for fixed d, Aff(d,L,1) is a polynomial of degree d in L. This formula is a generalization of the formula for the number of convex regions determined by L

hyperplanes in general position in $R^{(d)}$ (see Reference 7). The regions enumerated above are determined by L pairs of parallel hyperplanes in $R^{(d)}$.

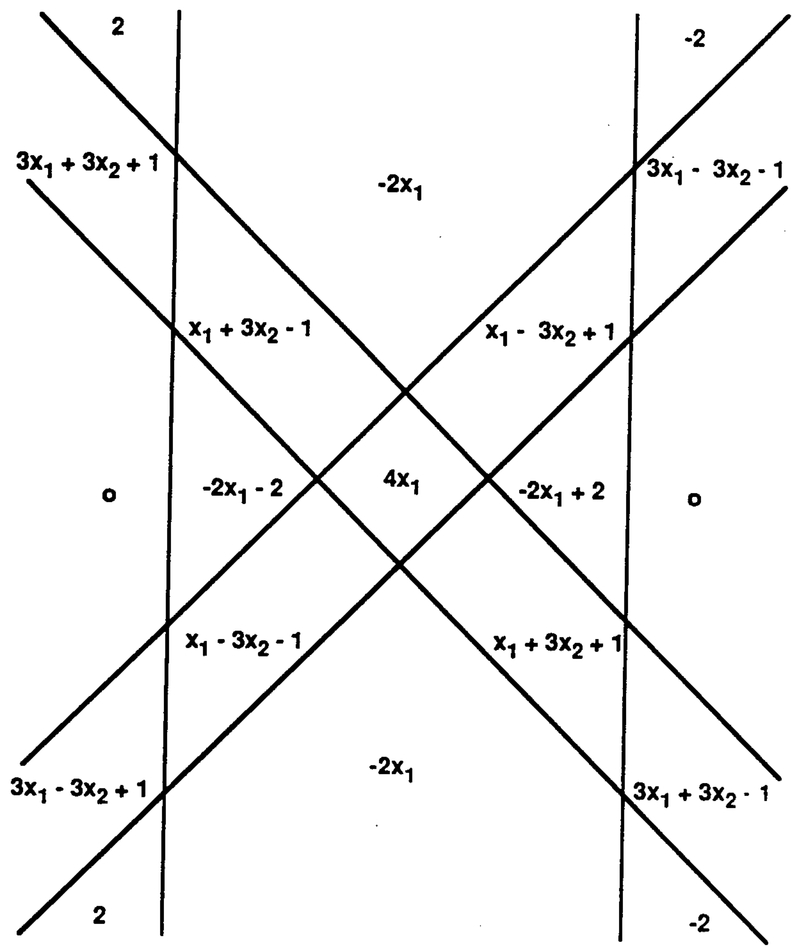

FIGURE 2. Decomposition of R^2 into 19 Affine Regions by (2,3,1)-Network Before Final Squashing.

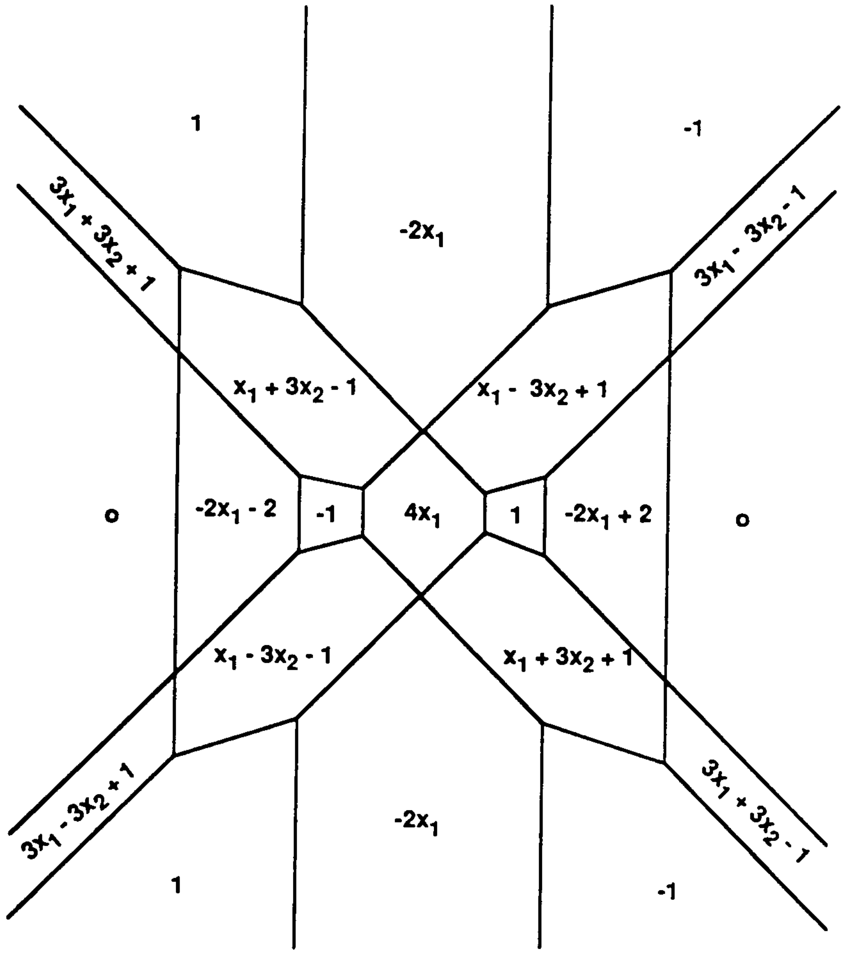

FIGURE 3. Decomposition of R^2 into 21 Affine Regions by (2,3,1)-Network After Final Squashing.

Henceforth, we denote members of the input space by $x_i = (x_{1,i}, x_{2,i}, ..., x_{d,i})^T$, outputs from the first hidden layer by $u_i = (u_{1,i}, y_{2,i}, ..., y_{L,i})^T$, and members of the output space by $y_i = (y_{1,i}, y_{2,i}, ..., y_{m,i})^T$. For the remainder of this section we discuss only (d,L,m)-networks.

We let $f : R^{(d)} \rightarrow R^{(L)}$ denote the mapping $x \rightarrow u$ realized at the output of the hidden layer. The jth coordinate of the output of the hidden layer defines a mapping $f_j : R^{(d)} \rightarrow R^{(1)}$. This mapping is determined by the jth row of the first weight matrix A. Let a_j denote the L-vector determined by the first L coordinates of the jth row of A, and let α_j be the (L+1)st coordinate in the jth row of A. Then

$$f_j(x) = p\left(a_j x + \alpha_j\right) \quad .$$

This function is piecewise affine with three regions:

$$f_j(x) = \left. \begin{array}{ll} -1 & \textit{for } x \,\varepsilon\, H_- \\ a_j x + \alpha_j & \textit{for } x \,\varepsilon\, H_o \\ 1 & \textit{for } x \,\varepsilon\, H_+ \end{array} \right\} \ .$$

H_-, H_+ are the half-spaces where $a_j x + \alpha_j$ is less than -1, greater than +1, respectively, and H_0 is the 'slab' between them. One can consider the single neuron mapping as projecting the slab onto an interval and the half-spaces onto its endpoints.

The following lemma illustrates how the (INT) relation can be altered by PLN mappings. Two neurons are sufficient for switching points between Ext(X) and Int(X) at the hidden layer. We let U denote the set f(X) of N outputs at the hidden layer. Lemma 1 says that an interior point x_1 of X can become an exterior point u_1 of U, while guaranteeing that any additional d-1 points can be placed in the interior of U. It should be noted that three points are required in Ext(U), since u-space is two-dimensional. This means that X must have at least d+2 points.

LEMMA 1

Suppose $X = \{x_1, x_2, ..., x_{d+2}\}$ is a (d+2)-set in general position in $R^{(d)}$, and x_1 ' Int(X). Then there exist weights for a (d,2,m)-PLN for which $f(x_1)$ ´ Ext(U) and $f(x_i)$ ´ Int(U), $2 \leq i \leq d$. That is, in two-dimensional u-space, the output of the hidden layer, Ext(U) is the triangle $\{f(x_1), f(x_{d+1}), f(x_{d+2})\}$

Outline of Proof

Let

$$X' = \{x_1, x_2, ..., x_d\} \ .$$

The mapping $f : R^{(d)} \rightarrow R^{(2)}$ is defined by

$$f(x) = (f_1(x), f_2(x))^T$$

where

$$f_j(x) = p(a_j x + \alpha_j) \ , \ 1 \leq j \leq 2 \ .$$

The (d-1)-simplex generated by X' lies in a unique hyperplane G_0. Since x_1 ´ Int(X), G_0 separates x_{d+1} and x_{d+2}. Let G_j denote the hyperplane through x_{d+j}, which is parallel to

G_0, and let g_j be the affine functional which is -1 on G_0 and 1 on G_j, for j = 1,2. Then the images of the x_i's under the mapping $g : x \rightarrow (g_1(x), g_2(x))^T = v$ are given by

$$v_i = g(x_i) = \begin{matrix} (1,-1)^T \ for\ i = d+1 \\ (-1,-1)^T \ for\ i \le i \le d \\ (-1,1)^T \ for\ i = d+2 \end{matrix}$$

The desired mapping f, which must place each u_i, $2 \le i \le d$, inside the triangle formed by u_1, u_{d+1}, and u_{d+2}, is obtained by perturbing the mapping g. In so doing the points x_i, $2 \le i \le d$, are clustered at (-1, -1) inside the triangle.

The complete proof, including the algebraic details of the construction of f, is presented in Appendix A.

4. ORDER-MODIFYING MAPPINGS

In this section two theorems are proved and two examples of PLN mappings are constructed. The theorems pertain to (d,d,d) and (d,d,d,d) PLNs. Theorem 2 establishes a new upper bound on $N_{max}(d,d,d)$, while Theorem 3 constructs (d,d,d,d) deformations of 2d+1 points in $R^{(d)}$, which cannot be realized by (d, d, d) networks. The two examples are included to illustrate how planar order relations can be modified by (2,2,2,2) PLNs.

Order is a fundamental algebraic and geometric concept. One of the better known linearly ordered sets is $R^{(1)}$ with the usual 'less than' order relation denoted $\le$. As was pointed out in Section 2, the mapping capabilities of PLN networks arise in a fundamental way from destruction of the (INT) relationship in finite subsets of Euclidean spaces $R^{(d)}$. For d=1 the (INT) relationship is based upon $\le$. For x a member of a finite subset X of $R^{(1)}$, x ´ Int(X) if, and only if, min(X) < x < max(X). This dependence of (INT) upon order in $R^{(1)}$ suggests the possibility of generalizations to higher dimensions. In this section the notion of order will be generalized from $R^{(1)}$ to $R^{(d)}$, as developed in Reference 13.

A partially ordered set (poset) is a set X endowed with a partial order P satisfying

(Ord 1) x P x

(Ord 2) if x P y and y P x, then x = y

(Ord 3) if x P y and y P z, then x P z.

A linear order satisfies the additional requirement

(Ord 4) for all x and y, either x P y or y P x.

An example of a poset, which is not linearly ordered, is $R^{(2)}$ with the order P, defined by:

$(x_1, x_2)\, P (y_1, y_2)$ whenever $x_1 \leq y_1$ and $x_2 \leq y_2$.

In this ordering the points (0,1) and (1,0) are not comparable so (Ord 4) does not hold.

Redefining the ≤ order in $R^{(1)}$, using signs lead to the following natural algebraic definition of higher dimensional order.

Suppose $T = (x_1, x_2, \ldots, x_{d+1})$ is a (d+1)-tuple in $R^{(d)}$, then T is called negative, degenerate or positive depending upon the value of the determinant of M(T), where

$$M(T) = \begin{bmatrix} 1 & 1 & & 1 \\ & & \cdots & \\ x_1 & x_2 & & x_{d+1} \end{bmatrix}_{(d+1)X(d+1)}$$

T is negative if det(M(T)) is negative

T is degenerate if det(M(T)) is zero

T is positive if det(M(T)) is positive.

For d = 1,

$$T = (x_1, x_2)$$

$$M(T) = \begin{bmatrix} 1 & 1 \\ x_1 & x_2 \end{bmatrix}$$

$$\det(M(T)) = x_2 - x_1$$

Thus, (x_1, x_2) is positive provided $x_1 < x_2$ as desired.

The three subfamilies into which the family of (d+1)-tuples from $R^{(d)}$ is decomposed by generalized order can be characterized as follows. The negative and positive (d+1)-tuples are oriented (ordered) d-simplexes. The degenerate (d+1)-tuples arise from sets which are not in general position in $R^{(d)}$. That is, T is degenerate provided there exists some hyperplane in $R^{(d)}$, which contains the set $\{x_1, x_2, ..., x_{d+1}\}$. It should be noted that permuting the coordinates of a degenerate T preserves degeneracy, while non-degenerate (d+1)-tuples alternate between negative and positive when pairs x_i, x_j are interchanged.

In $R^{(1)}$ the pair $T = (x_1, x_2)$ is positive whenever one passes to the right (in the usual graphic representation of the real line) when going from x_1 to x_2. Likewise in $R^{(2)}$, the triple $T = (x_1, x_2, x_3)$ is positive whenever one moves in a counter-clockwise direction about C when passing from x_1 to x_2 to x_3 to x_1, where $C = \mathrm{Conv}(\{x_1, x_2, x_3\})$.

Generalized order can be employed to categorize finite subsets in $R^{(d)}$. Given a k-subset $X = \{x_1, x_2, ..., x_k\}$ of $R^{(d)}$, each (d+1)-subset receives a label -, 0, or + depending upon its order. Of course, these labels change if the subscripts on the x's are changed. For a fixed labeling of the x's one obtains a mapping from the family of (d+1)-subsets of X to the set {-, 0, +}. The equivalence classes of mappings which are invariant under permutation of the k subscripts are called the unlabeled d-dimensional order types (Reference 13). The INT relation can also be utilized to define order by assigning the symbol Y or N to the pair (x, S) when x ´ Int(S) or $x \notin$ Int(S), respectively for all x ´ X and S £ X.

The following two theorems relate PLN mapping capabilities to generalized order properties of sets of inputs and outputs. The basic idea is that mapping capabilities as measured by $N_{max}(L^*)$ are related to the extent that order can be jumbled by an L^* network mapping. Theorem 1 says that one cannot quite turn a particular (2d+1)-subset of $R^{(d)}$ inside out with a (d,d,d)-PLN. On the other hand Theorem 2 demonstrates a way to do this with a (d,d,d,d)-PLN.

THEOREM 1. $N_{max}(d,d,d) \leq 2d$

Proof. It is sufficient to exhibit a set of 2d+1 input/output pairs that cannot be accommodated by a (d,d,d)-PLN. The following is such a set. Let $x_1, x_2, ..., x_{d+1}$ be the d+1 vertices of a d-simplex S in the interior of $I^{(d)}$, and let $x_{d+2}, x_{d+3}, ..., x_{2d+1}$ be d points chosen in the interior of S so that the entire set $X = \{x_1, x_2, ..., x_{2d+1}\}$ is in general position. The outputs, which also lie in $R^{(d)}$, form a permutation of the inputs. Specifically

$$\left.\begin{array}{rll} & x_i \quad for & i=1 \\ y_i = & x_{i+d} \quad for & 2 \leq i \leq d+1 \\ & x_{i-d} \quad for & d+2 \leq i \leq 2d+1 \end{array}\right\}$$

The d interior points of X must be interchanged with d of the vertices (extreme points) of X. This amounts to nearly turning the simplex inside out. We now show that this is not possible with one hidden layer.

Consider the values of the 2d+1 points in a fixed coordinate (neuron) of the hidden layer (u-space). At least two of the d+1 exterior inputs of X must assume extreme values at the fixed u-coordinate. Since the mapping from the hidden layer to the output space is 1-1 on the set of interest, all exterior points of the convex hull of the image of X in u-space must also be extreme points in the output space. In particular, at least two of the exterior inputs must be vertices in the output set. Since only one of the input vertices goes to an output vertex, namely, x_1, a contradiction arises. Thus, no mapping sending x_i into y_i, for $1 \leq i \leq 2d+1$, exists.

The following two examples of (2,2,2,2)-PLN mappings illustrate how a second hidden layer can facilitate order modification. Table 4 and Figure 4 show the 5 sets of X and Y inputs and outputs, respectively, for Example 4. Ext(X) = $\{x_1, x_2, x_3\}$ with x_4 and x_5 lying inside the 2-simplex. The line joining x_4 and x_5 cuts the faces $\{x_1, x_3\}$ and $\{x_2, x_3\}$ of the 2-simplex. The output set Y also consists of a 2-simplex $\{y_1, y_2, y_3\}$, $y_i = x_i$, $1 \leq i \leq 3$, with two interior points $\{y_4, y_5\}$. The line joining y_4 and y_5 also cuts faces $\{y_1, y_3\}$ and $\{y_2, y_3\}$. However the triples $\{x_1, x_2, x_3\}$ and $\{x_3, x_4, x_5\}$ have the same sign, while $\{y_1, y_2, y_3\}$ and $\{y_3, y_4, y_5\}$ have opposite signs.

TABLE 4. Inputs and Outputs for Example 4.

i	x_i^T	y_i^T
1	(-0.5000, -0.5000)	(-0.5000, -0.5000)
2	(0.5000, -0.5000)	(0.5000, -0.5000)
3	(0.0000, 0.5000)	(0.0000, 0.5000)
4	(-0.1625, -0.1250)	(0.2724, -0.1696)
5	(0.2250, -0.2500)	(-0.1427, -0.2143)

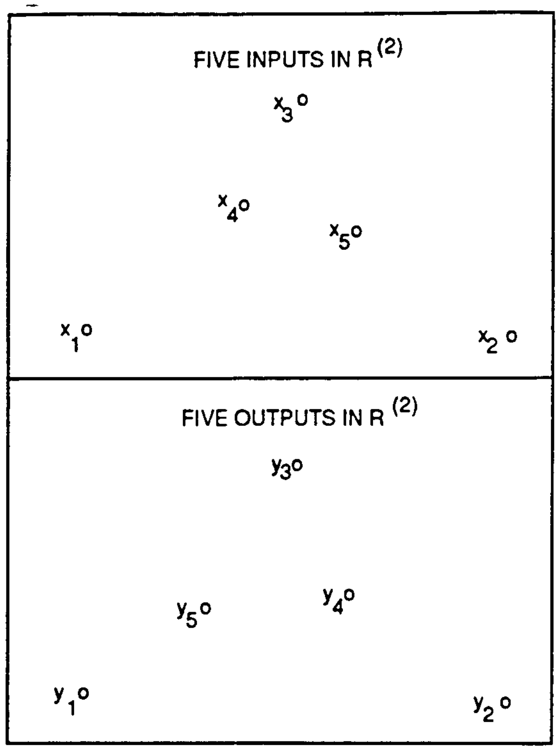

FIGURE 4. Five Inputs and Five Outputs for Example 4.

Table 5 and Figure 5 show the input and output sets of Example 5. The labeled order type (pattern of signs) of X is the same as for Example 4. The outputs for Example 5 are, however, different; Ext(Y) = $\{y_3, y_4, y_5\}$ with y_1 and y_2 lying inside the simplex. As in Example 4, the triples $\{x_1, x_2, x_3\}$ and $\{x_3, x_4, x_5\}$ have the same sign; while $\{y_1, y_2, y_3\}$ and $\{y_3, y_4, y_5\}$ have opposite signs. The interior line through y_1 and y_2 cuts the faces $\{y_3, y_4)$ and $\{y_4, y_5)$.

TABLE 5. Inputs and Outputs for Example 5.

i	x_i^T	y_i^T
1	(-0.5000, -0.5000)	(0.0784, -0.4722)
2	(0.5000, -0.5000)	(0.0334, -0.3808)
3	(0.0000, 0.5000)	(0.0000, 0.5000)
4	(-0.1000, -0.1000)	(-0.4999, -0.5000)
5	(0.1000, -0.1000)	(0.5000, -0.5000)

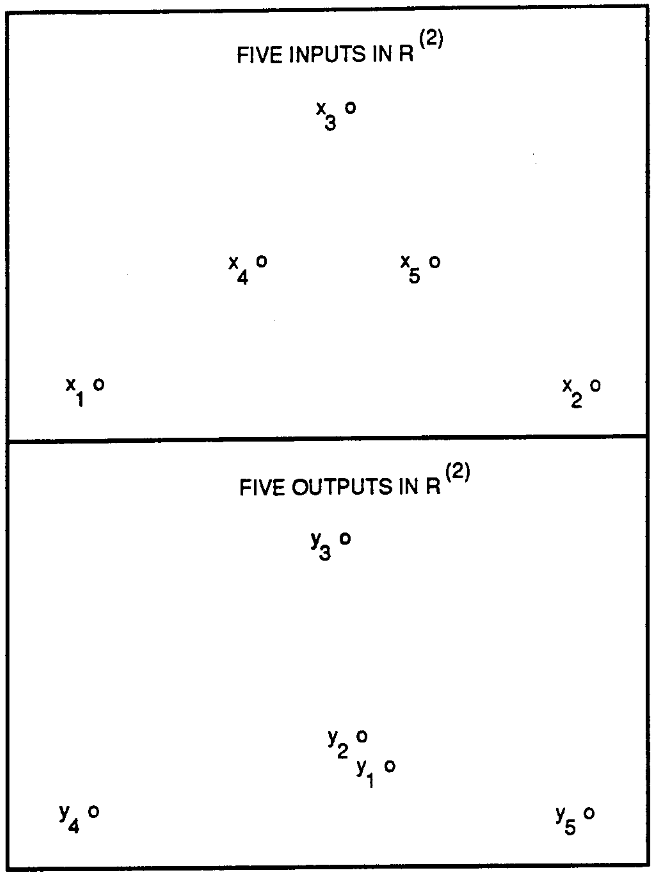

FIGURE 5. Five Inputs and Five Outputs for Example 5.

Tables 6 and 7 contain the weight matrices for the two examples; while Tables 8 and 9 show the u-space and v-space outputs at the hidden layers.

TABLE 6. Three Weight Matrices for Example 4.

$$A_1 = \begin{bmatrix} 1.4286 & -1.2857 & -0.3571 \\ 0.7619 & -3.3524 & -0.2952 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0.5833 & -1.1667 & 0.4167 \\ 3.1500 & -4.3750 & 2.2250 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} -0.4286 & 0.6786 & -0.2500 \\ 0.8571 & -0.3571 & 0.0000 \end{bmatrix}$$

TABLE 7. Three Weight Matrices for Example 5.

$$A_1 = \begin{bmatrix} 5.000 & -1.6667 & 0.0000 \\ 2.2727 & 0.4545 & 0.1818 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 2.4545 & -2.5455 & 0.3485 \\ 3.1818 & -3.2727 & 0.2273 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} -2.0706 & 2.3206 & 0.2500 \\ -2.5810 & 2.0810 & 0.0000 \end{bmatrix}$$

TABLE 8. Intermediate Outputs for Example 4.

i	u_i^T	v_i^T
1	(-0.4286, 1.0000)	(-1.0000, -1.0000)
2	(1.0000, 1.0000)	(-0.1667, 1.0000)
3	(-1.0000, -1.0000)	(1.0000, 1.0000)
4	(-0.4285, 0.0000)	(0.1668, 0.8752)
5	(0.2858, 0.7143)	(-0.2500, 0.0002)

TABLE 9. Intermediate Outputs for Example 5.

i	u_i^T	v_i^T
1	(-1.0000, -1.0000)	(0.4395, 0.3182)
2	(1.0000, 1.0000)	(0.2575, 0.1364)
3	(-0.8334, 0.4091)	(-1.0000, -1.0000)
4	(-0.3333, -0.0909)	(-0.2382, -0.5357)
5	(0.6667, 0.3636)	(1.0000, 1.0000)

Example 5 utilizes input/output pairs, which for d = 2, are similar to the sets employed in the proof of Theorem 2. The argument of Theorem 2 applies to Example 5. Thus, the action of the (2,2,2,2)-PLN mapping on X, of Example 5, cannot be realized with a (2,2,2)-PLN.

The following theorem shows that generalizations of the (2,2,2,2) mapping of Example 5 exist for all (d,d,d,d)-PLNs.

THEOREM 2

Suppose X is a (2d+1)-set in general position in $R^{(d)}$, X = S ¡ T, |S| = d, |T| = d+1, and S is a facet in Ext(X). Suppose further that no line joining two points of T is parallel to the hyperplane through S. Then there is a weight assignment W for a (d,d,d,d)-PLN for which $F_W(S) = \mathrm{Int}(F_W(X))$, $F_W(T) = \mathrm{Ext}(F_W(X))$, and $F_W(X)$ lies in the interior of $I^{(d)}$.

Remarks

This theorem says that the set X, consisting of a d-simplex and d-interior points, can almost be turned inside out. That is, in the output space $I^{(d)}$, d of the exterior points become interior while the d interior points become exterior. The purpose of placing the output set within the interior of $I^{(d)}$ is to achieve the result without benefit of the squashing function at the output layer. It should be emphasized that Theorem 2 does not say that any mapping between (2d+1)-sets X and Y, each consisting of a d-simplex and d-interior points, can be achieved by a (2,2,2,2)-PLN. The theorem only guarantees the certain Ys can be achieved, which cannot be handled with (2,2,2)-PLNs.

5. SUMMARY

The feed-forward layered neural network has great potential for fast computation of discriminant functions and other transformations required in image processing and pattern recognition. Network design and weight assignment are two of the important tasks

arising in neural network applications. The results presented here pertain to mapping construction and capabilities for layered networks with piecewise linear neuron transfer function.

The main focus of this work is two-fold. First it is shown that a certain type of mapping in d-dimensional Euclidean space cannot be achieved by a (d,d,d)-PLN (piecewise linear network). The mapping of interest involves turning the simplex inside out in Euclidean d-space. It is then shown that such mappings can be achieved by a (d,d,d,d)-PLN. The importance of these results lies in the methodology of the proofs as well as the construction techniques, rather than in the treatment of the particular mapping in d-space. It is also shown that two hidden neurons are sufficient for moving a point from the interior of a set to its exterior. It is this ability to disturb the order properties of Euclidean sets, which fosters the mapping complexity of piecewise linear networks.

REFERENCES

1. Naval Weapons Center. *Equation of Learning and Capacity of Layered Neural Networks,* by Jorge M. Martin. China Lake, Calif., NWC, May 1989. (NWC TP 7013, publication UNCLASSIFIED.)

2. ___________. *On the Interpolation Properties of Feedforward Layered Neural Networks,* by Jorge M. Maritin. China Lake, Calif., NWC, October 1990. (NWC TP 7094, publication UNCLASSIFIED.)

3. ___________. *Low Sensitivity Interpolation Using Feed-Forward Neural Networks With One Hidden Layer,* by Jorge M. Maritin. China Lake, Calif., NWC, December 1991. (NWC TP 7191, publication UNCLASSIFIED.)

4. Naval Air Warfare Center Weapons Division. *Piecewise Linear Neural Networks,* by W. O. Alltop. China Lake, Calif., NAWCWPNS, August 1992. (NAWCWPNS TP 8071, publication UNCLASSIFIED.)

5. B. Noble. *Applied Linear Algebra.* Engelwood Cliffs, New Jersey, Prentice-Hall, 1969.

6. F. A. Valentine. *Convex Sets.* New York, McGraw-Hill, 1964.

7. H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* Berlin, Springer-Verlag, 1987.

8. F. P. Preparata and M. I. Shamos. *Computational Geometry.* New York, Springer-Verlag, 1985.

9. E. B. Baum. "On the Capabilities of Multi-Layer Perceptrons," *J. Complexity,* Vol. 4 (1988). Pp. 193-215.

10. R. P. Lippmann. "An Introduction to Computing With Neural Nets," *IEEE ASSP Mag.* (April 1987). Pp. 4-22.

11. Naval Weapons Center. *Neuron Requirements for Classifications,* by W. O. Alltop. China Lake, Calif., NWC, January 1991. (NWC TP 7106, publication UNCLASSIFIED.)

12. E. D. Sontag. Feedforward Nets for Interpolation and Classification," *J. Comput. System Sci.,* Vol. 45 (August 1992). Pp. 20-48.

13. J. E. Goodman and R. Pollack. "Multi-Dimensional Sorting," *SIAM J. Computing,* Vol. 12, No.3 (1983). Pp. 484-507.

Appendix A

PROOF OF LEMMA 1

LEMMA 1

Suppose $X = \{x_1, x_2, ..., x_{d+2}\}$ is a (d+2)-set in general position in $R^{(d)}$, and x_1 ´ Int(X). Then there exist weights for a (d,2,m)-PLN for which $f(x_1)$ ´ Ext(U) and $f(x_i)$ ´ Int(U), $2 \leq i \leq d$. That is, in two-dimensional u-space, the output of the hidden layer, Ext(U), is the triangle $\{f(x_1), f(x_{d+1}), f(x_{d+2})\}$.

Proof

Let

$$X' = \{x_1, x_2, ..., x_d\} \ .$$

The mapping $f : R^{(d)} \rightarrow R^{(2)}$ is defined by

$$f(x) = (f_1(x), f_2(x))^T$$

where

$$f_j(x) = p(a_j x + \alpha_j) \ , 1 \leq j \leq 2 \ .$$

The desired mapping f must place each u_i, $2 \leq i \leq d$, inside the triangle formed by u_{d+1}, u_1, and u_{d+2}, where

$$u_{d+1} = (1, -1)^T$$

$$u_1 = (-1, -1)^T$$

$$u_{d+2} = (-1, 1)^T \ .$$

For the sake of clarity f will be defined algebraically.

There exists a unique unit vector c' and scalar d' satisfying

$$z'_i = c' x_i + d' \left.\begin{aligned} &< 0 \; for \; i = d+1 \\ &= 0 \; for \; 1 \le i \le d \\ &> 0 \; for \; i = d+2 \end{aligned}\right\}$$

Next let $c_0 = \gamma c'$ and $d_0 = \gamma d'$, where

$$\gamma = 1 / \min[-z'_{d+1}, z'_{d+2}]$$

This gives

$$z_i = c_0 x_i + d_0 \left.\begin{aligned} &= -1 - \delta_1 \; for \; i = d+1 \\ &= 0 \qquad for \; 1 \le i \le d \\ &= 1 + \delta_2 \; for \; i = d+2 \end{aligned}\right\}$$

where $\delta_1 \ge 0$ and $\delta_2 \ge 0$. There exists a neighborhood N_0 of c_0 satisfying the following:

for all c ´ N_0,

$|cx_i - c_0 x_i| \le K$ for $1 \le i \le d+2$,

where

$$K = \frac{1}{3} \min\left[|c_0 x_1 - c_0 x_{d+1}|, |c_0 x_1 - c_0 x_{d+2}| \right] \quad .$$

This choice of N_0 guarantees the existence of c'' ´ N_0 satisfying

$$c'' x_{d+1} < c'' x_1 < c'' x_2 < \ldots < c'' x_d < c'' x_{d+2} \quad .$$

There also exists a neighborhood N of c'', which is contained in N_0, and satisfies

$$cx_{d+1} < cx_1 < cx_2 < \ldots < cx_d < cx_{d+2}$$

for all c ´ N. Finally we chose two linearly independent vectors c_1, c_2 in N. The vectors a_j and the scalars α_j, j = 1,2, are determined by the c_j, $0 \leq j \leq 2$, and constants σ_j, τ_j, j = 1,2. In particular

$$a_j = \sigma_j c_0 + \tau_j c_j$$

and

$$\alpha_j = \sigma_j d_0 - 1 - \tau_j c_j x_1 \quad .$$

For all choices of σ_j, τ_j, j = 1,2,

$$a_j x_1 + \alpha_j = -1 \quad ,$$

and

$$a_j x_i + \alpha_j = -1 + \tau_j c_j (x_i - x_1) \quad , \text{ for } 2 \leq i \leq d.$$

Next we let

$$\tau_j = \frac{\varepsilon}{c_j(x_d - x_1)}, j = 1,2$$

where 0 < ´ < 1. This guarantees that

$$-1 \leq a_j x_i + \alpha_j < -1 + ´ < 0$$

for j - 1,2, and $2 \leq i \leq d$.

For the remaining points x_{d+1}, x_{d+2}, we have

$$a_j x_{d+1} + \alpha_j = -1 - \sigma_j(1 + \delta_1) - \tau_j\, c_j\, (x_1 - x_{d+1})$$

$$a_j x_{d+2} + \alpha_j = -1 + \sigma_j(1 + \delta_2) + \tau_j\, c_j\, (x_{d+2} - x_1)$$

$$\text{for } j = 1,2.$$

The values of σ_1, σ_2, are chosen so as to move $a_j x_i + \alpha_j$, $j = 1,2$, $i = d+1, d+2$, beyond the thresholds -1, +1:

$$\sigma_1 = -\max\,[M_1, M_2]$$

$$\sigma_2 = \max\,[M_3, 0]$$

where

$$M_1 = \frac{2 + \tau_1 c_1 (x_1 - x_{d+1})}{1 + \delta_1}$$

$$M_2 = \frac{\tau_1 c_1 (x_{d+2} - x_1)}{1 + \delta_2}$$

$$M_3 = \frac{2 - \tau_2 c_2 (x_{d+2} - x_1)}{1 + \delta_2} \quad .$$

With these assignments of a_j, α_j, the following inequalities hold

$$a_1 x_{d+1} + \alpha_1 \geq 1 \ , \ a_2 x_{d+1} + \alpha_2 \leq -1 \quad ,$$

$$a_1 x_{d+2} + \alpha_1 \leq -1 \ , \ a_2 x_{d+2} + \alpha_2 \geq 1 \ .$$

The two-dimensional outputs u_i at the hidden layer are given by

$$u_i = (u_{1,i}, u_{2,i})^T \quad ,$$

where

$$u_{j,i} = p(a_j x_i + \alpha_j) \quad .$$

Table A-1 shows the coordinates of the u_i, $1 \le i \le d+2$.

The choice of ´ is critical in positioning the u_i, $2 \le i \le d$. For 0 < ´ < 2, all u_i lie inside the square with vertices (±1, ±1). In order to guarantee that the u_i lie in the triangle formed by u_{d+1}, u_1, u_{d+2}, one must also require ´ < 1. As ´ approaches 0, the u_i all approach u_1.

TABLE A-1. Coordinates of d+2 Points in the u-Plane.

i	$u_{1,i}$	$u_{2,i}$
1	-1	-1
2	-1+´$_{1,2}$	-1+´$_{2,2}$
3	-1+´$_{1,3}$	-1+´$_{2,3}$
⋮	⋮	⋮
d-1	-1+´$_{1,d-1}$	-1+´$_{2,d-1}$
d	-1+´	-1+´
d+1	1	-1
d+2	-1	1

0 < ´$_{j,1}$ < ´$_{j,3}$ < ... < ´$_{j,d-1}$ < ´ < 1

for j = 1,2

Appendix B

PROOF OF THEOREM 2

Throughout this appendix we assume that $d \geq 2$. Lemma 2 establishes the following useful fact; the minimum member of a set of d+2 real numbers can be placed anywhere inside the convex hull (d-simplex) of the remaining d+1 members by a (1,d,d)-PLN.

LEMMA 2

Let V denote the d-simplex in $R^{(d)}$ with vertices v_i, $2 \leq i \leq d+2$, where

$$v_2 = (-1,-1,-1,\dots,-1,-1)^T$$

$$v_3 = (1,-1,-1,\dots,-1,-1)^T$$

$$v_4 = (1,1,-1,\dots,-1,-1)^T$$

$$v_5 = (1,1,1,\dots,-1,-1)^T$$

$$\vdots \qquad \vdots$$

$$v_{d+1} = (1,1,1,\dots,1,-1)^T$$

$$v_{d+2} = (1,1,1,\dots,1,1)^T$$

and let v_1 be any point inside V. If $z_1, z_2, \dots, z_{d+2}$ are real numbers satisfying

$$z_1 < z_2 < \dots < z_{d+2} \quad ,$$

then there exist weights for a (1,d,d)-PLN which map z_i into v_i, $1 \leq i \leq d+2$.

Proof

The first layer weights a_j, α_j are given by

$$a_1 = \frac{2}{z_2 - z_1}$$

$$\alpha_1 = \frac{z_1 + z_2}{z_1 - z_2}$$

$$\left.\begin{aligned} a_j &= \frac{2}{z_{j+2} - z_1} \\ \alpha_j &= \frac{z_1 + z_{j+2}}{z_1 - z_{j+2}} \end{aligned}\right\} for\ 2 \le j \le d \quad .$$

The mapping $z_i \rightarrow u_i$ from input space to the output of the hidden layer is defined by

$$u_{j,i} = p(a_j z_i + \alpha_j) ,$$

giving

$$u_1 = (-1,-1,-1,-1,\ldots,-1,-1)^T$$

$$u_2 = (1,u_{2,2},u_{3,2},u_{4,2},\ldots,u_{d-1,2},u_{d,2})^T$$

$$u_3 = (1,u_{2,3},u_{3,3},u_{4,3},\ldots,u_{d-1,3},u_{d,3})^T$$

$$u_4 = (1,1,u_{3,4},u_{4,4},\ldots,u_{d-1,4},u_{d,4})^T$$

$$u_5 = (1,1,1,u_{4,5},\ldots,u_{d-1,5},u_{d,5})^T$$

$$u_6 = (1,1,1,1,..,u_{d-1,6},u_{d,6})^T$$

$$\vdots$$

$$u_{d+1} = (1,1,1,1,\ldots,1,u_{d,d+1})^T$$

$$u_{d+2} = (1,1,1,1,\ldots,1,1)^T$$

From the monotinicity of the z_i's it also follows that

$$-1 < u_{j,2} < u_{j,3} < u_{j,4} < \ldots < u_{j,j} < u_{j,j+1} < 1$$

for $2 \leq j \leq d$, i.e. all rows of the matrix of u_i's are monotone.

In order to realize the specific positions of 1's and -1's in the u_i's, the first layer weights are uniquely determined as shown above. Slight perturbations of the z_i's, which preserve montonicity, will result in slight perturbations of the a_j's and α_j's. These perturbed weights produce the same pattern of +1's and -1's, while perturbing the remaining $u_{j,i}$'s slightly.

The second layer weights $b_{j,k}$, β_j, are given by

$$b_{1,1} = -1 - \frac{1}{2}v_{1,1} + \frac{2u_{2,2}+2}{u_{2,2}-u_{2,3}}$$

$$b_{1,2} = \frac{4}{u_{2,3} - u_{2,2}}$$

$$\beta_1 = -1 + \frac{1}{2} v_{1,1} + \frac{2u_{2,2} - 2}{u_{2,2} - u_{2,3}}$$

$$\left.\begin{aligned} b_{j,1} &= 1 - \frac{1}{2} v_{j,1} - \frac{4}{1 - u_{j,j+1}} \\ b_{j,j} &= \frac{4}{1 - u_{j,j+1}} \\ \beta_j &= 1 + \frac{1}{2} v_{j,1} \end{aligned}\right\} \textit{for } 2 \le j \le d$$

and all other $b_{j,k}$'s are zero. The images of the u_i's under the affine mapping $u \rightarrow Bu + \beta$, before squashing, are shown below, where

$$B = \left[b_1^T, b_2^T, \ldots, b_d^T\right]^T = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_d \end{bmatrix}$$

$$b_j = \left[b_{j,1} b_{j,2}, \ldots, b_{j,d}\right]$$

and

$$\beta = (\beta_1, \beta_2, \ldots, \beta_d)^T$$

$$b_j u_1 + \beta_j = v_{j,1} \textit{ for } 1 \le j \le d$$

$$b_1 u_2 + \beta_1 = -2$$

$$b_1 u_3 + \beta_1 = 2$$

$$b_1 u_i + \beta_1 = 2 + \frac{4(1 - u_{2,3})}{u_{2,3} - u_{2,2}} > 2$$
$$\text{for } 4 \le i \le d+2$$

$$b_j u_2 + \beta_j = -2 + \frac{4(u_{j,2} - u_{j,j+2})}{1 - u_{j,j+1}} < -2$$
$$\text{for } 2 \le j \le d$$

$$b_j u_i + \beta_j = -2 + \frac{4(u_{j,i} - u_{j,j+1})}{1 - u_{j,j+1}} < -2$$
$$\text{for } i \le j+1$$
$$\text{and } 2 \le j \le d$$

$$b_j u_i + \beta_j = 2 + \frac{4(u_{j,i} - 1)}{1 - u_{j,j+1}} = 2$$
$$\text{for } i \ge j+2$$
$$\text{and } 2 \le j \le d$$

After squashing (application of the function p) we have $p(Bu_i + \beta) = v_i$. It is important to note that the values before squashing satisfy

$$|b_j u_i + \beta_j| \ge 2$$

for all j when $i \ge 2$. This is helpful when considering small perturbations in the data. Suppose that u_i' lies in a small neighborhood of u_i, for $2 \le i \le d+2$. Then $p(Bu_i' + \beta) = v_i$, for all i. Moreover, if u_1' is close to u_1, then $v_1' = p(Bu_1' + \beta)$ will lie in a small neighborhood of v_1. Indeed a sufficiently small neightorhood of u_1 can be mapped into a neighborhood of v_1, which lies in the interior of V.

THEOREM 2

Suppose X is a (2d+1)-set in general position in $R^{(d)}$, $X = S \cup T$, $|S| = d$, $|T| = d+1$, and S is a facet in Ext(X). Suppose further that no line joining two points of T is parallel to the hyperplane through S. Then there is a weight assignment W for a (d,d,d,d)-PLN for which $F_W(S)$ lies inside the interior of $Conv(F_W(T))$. Moreover $F_W(T)$ lies in the interior of the unit cube.

Outline of Proof

The proof employs an intermediate set of weights $(A^{(0)}, \alpha^{(0)}, B, \beta, I_d, 0_d)$ as well the final weights $W = (A, \alpha, B, \beta, C, \gamma)$. Here I_d is the d by d identity matrix and 0_d is the d-vector of zeroes. The first set maps T to the d-simplex V while mapping all of S to the single point v_1 inside V. The first layer weights $(A^{(0)}, \alpha^{(0)})$ are perturbed slightly to obtain (A, α). The two-layer mapping (A, α, B, β) also sends T to V while mapping S into a cluster of points in a small neighborhood of v_1 lying entirely within the interior of V. The third layer weights (C, γ) just map the simplex V into the interior of $[-1, 1]^{(d)}$, so that squashing at the output layer is irrelevant. The inequalities satisfied by $b_j u_i + \beta_j$ allow the use of the second layer weights (B, β) in both mappings.

The linear functional $x \rightarrow hx$, which is constant on S, maps x_i into the $z_i^{(0)}$. The first layer weights $(A^{(0)}, \alpha^{(0)})$ are then determined by the vector $(z_d^{(0)}, z_{d+1}^{(0)}, ..., z_{2d+1}^{(0)})$. These determine the $u_i^{(0)}$'s which, together with v_1 determine the second layer (B, β). $\|B\|$, v_1, and the $z_i^{(0)}$'s are used to define a small neighborhood η_1 of h in the boundary ∂Sph of the unit sphere Sph in $R^{(d)}$. A suitable $h^{(1)}$ is selected in η_1 which defines the mapping $x_i \rightarrow z_i^{(1)}$. The $z_i^{(1)}$'s in turn determine a neighborhood η_2 of $h^{(1)}$ in ∂Sph. Finally a basis $h_1, h_2, ..., h_d$ of vectors is chosen from η_2. These functionals are used to define the first layer (A, α) of weights.

Proof

We let $S = \{x_1, x_2, ..., x_d\}$ and $T = \{x_{d+1}, x_{d+2}, ..., x_{2d+1}\}$. There exists a unique hyperplane H through S. Since S is a facet of X, the set T is not separated by H. Thus, there exists a unique unit vector $h^{(0)}$ and a unique scalar z satisfying

$$h^{(0)}x_i = z \text{ if } 1 \le i \le d$$

$$h^{(0)}x_i > z \text{ if } d+1 \le i \le 2d+1 \quad .$$

Letting $z_1^{(0)} = h^{(0)}x_i$, we have

$$z_1^{(0)} = z_2^{(0)} = \ldots = z_d^{(0)} < z_i^{(0)}$$

for $d+1 \leq i \leq 2d+1$. Moreover, since no line through two members of T is parallel to H, the $z_i^{(0)}$'s, must be distinct, for $d+1 \leq i \leq 2d+1$. Therefore, after relabeling (if necessary), we have

$$z_d^{(0)} < z_{d+1}^{(0)} < z_{d+2}^{(0)} < \cdots < z_{2d+1}^{(0)} \quad .$$

The $z_i^{(0)}$'s play the role of the z_i's in the preceding Lemma, after setting $z_i = z_{d+i-1}^{(0)}$, $1 \leq i \leq d+2$.

The intermediate first layer weights $(A^{(0)}, \alpha^{(0)})$ are given by

$$A^{(0)} = A_1^{(0)} H^{(0)}$$

$$\alpha^{(0)} = \left(\alpha_1^{(0)}, \alpha_2^{(0)}, \cdots, \alpha_d^{(0)}\right)^T$$

where

$$A_1^{(0)} = \begin{bmatrix} a_1^{(0)} & 0 & 0 & 0 & \cdots & 0 \\ 0 & a_2^{(0)} & 0 & 0 & \cdots & 0 \\ 0 & 0 & a_3^{(0)} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \vdots & a_d^{(0)} \end{bmatrix}_{d \, X \, d}$$

and

$$H^{(0)} = \begin{bmatrix} h^{(0)} \\ h^{(0)} \\ \vdots \\ h^{(0)} \end{bmatrix}_{d\,X\,d}$$

The $a_i^{(0)}$'s and $\alpha_i^{(0)}$'s are given by

$$a_1^{(0)} = \frac{2}{z_{d+1}^{(0)} - z_1^{(0)}}$$

$$\alpha_1^{(0)} = \frac{z_1^{(0)} + z_{d+1}^{(0)}}{z_1^{(0)} - z_{d+1}^{(0)}}$$

$$\left.\begin{aligned} a_j^{(0)} &= \frac{2}{z_{j+d+1}^{(0)} - z_1^{(0)}} \\ \alpha_j^{(0)} &= \frac{z_1^{(0)} + z_{j+d+1}^{(0)}}{z_1^{(0)} - z_{j+d+1}^{(0)}} \end{aligned}\right\} for\ 2 \le j \le d \quad .$$

The outputs $u_{j,i}^{(0)}$, for $1 \geq j \leq d$ and $d \leq i \leq 2d+1$, are defined by

$$u_{j,i}^{(0)} = p\left(a_j^{(0)} z_i^{(0)} + \alpha_j^{(0)}\right) \quad ,$$

and take the following form.

$$u_{d+1}^{(0)} = \left(1, u_{2,d+1}^{(0)}, u_{3,d+1}^{(0)}, \cdots, u_{d-1,d+1}^{(0)}, u_{d,d+1}^{(0)}\right)^T$$

$$u_{d+2}^{(0)} = \left(1, u_{2,d+2}^{(0)}, u_{3,d+2}^{(0)}, \cdots, u_{d-1,d+2}^{(0)}, u_{d,d+2}^{(0)}\right)^T$$

$$u_{d+3}^{(0)} = \left(1, 1, u_{3,d+3}^{(0)}, \cdots, u_{d-1,d+3}^{(0)}, u_{d,d+3}^{(0)}\right)^T$$

$$u_{d+4}^{(0)} = \left(1, 1, 1, \cdots, u_{d-1,d+4}^{(0)}, u_{d,d+4}^{(0)}\right)^T$$

$$\vdots \qquad \vdots$$

$$u_{2d}^{(0)} = \left(1, 1, 1, \cdots, 1, u_{d,2d}^{(0)}\right)^T$$

$$u_{2d+1}^{(0)} = (1, 1, 1, \cdots, 1, 1)^T$$

The outputs $u_{j,i}^{(0)}$ of the jth neuron are monotonic:

$$-1 < u_{j,d+1}^{(0)} < u_{j,d+2}^{(0)} < u_{j,d+3}^{(0)} < \ldots < u_{j,d+j}^{(0)} < 1 \quad .$$

The second layer of weights (B, β) is given by

$$B = \left[b_{j,i}\right]_{d\ X\ d}$$

$$\beta = (\beta_1, \beta_2, \ldots, \beta_d)^T$$

where

$$b_{1,1} = -1 - \frac{1}{2}v_{1,1} + \frac{2u_{2,d+1}^{(0)} + 2}{u_{2,d+1}^{(0)} - u_{2,d+2}^{(0)}}$$

$$b_{1,2} = \frac{4}{u_{2,d+2}^{(0)} - u_{2,d+1}^{(0)}}$$

$$\beta_1 = -1 + \frac{1}{2}v_{1,1} + \frac{2u_{2,d+1}^{(0)} - 2}{u_{2,d+1}^{(0)} - u_{2,d+2}^{(0)}}$$

$$\left.\begin{aligned} b_{j,1} &= 1 - \frac{1}{2}v_{j,1} - \frac{4}{1 - u_{j,d+j}^{(0)}} \\ b_{j,j} &= \frac{4}{1 - u_{j,d+j}^{(0)}} \\ \beta_j &= 1 + \frac{1}{2}v_{j,1} \end{aligned}\right\} \mathit{for}\ 2 \le j \le d$$

and all other $b_{j,k}$'s are zero. It should be noted that the expressions for the $b_{j,k}$ and β_j are identical to those in Lemma 2 with each $u_{j,i}$ replaced by $u_{j,i+d}^{(0)}$.

As in Lemma 2 images of the $u_i^{(0)}$'s under the mapping $u \rightarrow Bu + \beta$, before squashing, satisfy

$$\left| b_j u_i^{(0)} + \beta_j \right| \ge 2 \quad ,$$

for $1 \le j \le d$ and $d+1 \le i \le 2d+1$.

The neighborhood η_1 depends upon the following constants:

$$K_1 = \frac{r_v}{2\|B\|\sqrt{d}}$$

$$K_2 = \max_j \left[\left\|a_j^{(0)}\right\|\right] = \frac{2}{z_{d+1}^{(0)} - z_1^{(0)}}$$

$$K_3 = \frac{K_1}{K_2\left[1 + 2K_2 + 4K_2 z_{\max}^{(0)}\right]}$$

$$r_v = \min[1, R_v]$$

$$z_{\max}^{(0)} = \max\left[\left|z_i^{(0)}\right| : 1 \le i \le 2d+1\right]$$

where R_v is the radius of a sphere centered at v_1, which is contained entirely within the simplex V, and ||B|| is the norm of the matrix B.

We let η_1 be a neighborhood of $h^{(0)}$ in ∂Sph satsifying the following

$$\left|h' x_i - h^{(0)} x_i\right| < \delta_1 \text{ for all } h' \varepsilon \eta_1$$
$$\text{and } 1 \le i \le 2d+1,$$

where δ_1 is the minimum of the five quantities

$$\frac{1}{2}$$

$$\frac{1}{2}K_1$$

$$\frac{1}{4K_2}$$

$$\frac{1}{2}K_3$$

$$\frac{1}{3}\min\left[z_{i+1}^{(0)} - z_i^{(0)} : d \le i \le 2d\right] \quad .$$

Since η_1 contains $h^{(0)}$, which maps the set S into the single point z, η_1 must contain some $h^{(1)}$ satisfying

$$z_1^{(1)} < z_2^{(1)} < \ldots < z_d^{(1)} \quad ,$$

where $z_i^{(1)} = h^{(1)}x_i$. From the constraints on δ_1 we have, for $d \le i \le 2d$,

$$\begin{aligned} z_{i+1}^{(1)} - z_i^{(1)} &= z_{i+1}^{(1)} - z_{i+1}^{(0)} + z_{i+1}^{(0)} - z_i^{(0)} + z_i^{(0)} - z_i^{(1)} \\ &> \left(z_{i+1}^{(0)} - z_i^{(0)}\right) - \left|z_{i+1}^{(1)} - z_{i+1}^{(0)}\right| - \left|z_i^{(0)} - z_i^{(1)}\right| \\ &> \min\left[z_{i+1}^{(0)} - z_i^{(0)} : d \le i \le 2d\right] - 2\delta_1 \\ &> \frac{1}{3}\min\left[z_{i+1}^{(0)} - z_i^{(0)}\right] > 0 \quad . \end{aligned}$$

This guarantees that monotonicity of the $z_i^{(1)}$'s is maintained, i.e.

$$z_1^{(1)} < z_2^{(1)} < \cdots < z_{2d+1}^{(1)} \quad .$$

The neighborhood η_2 of $h^{(1)}$ in ∂Sph is chosen so that

$$\left|h' x_i - h^{(1)} x_i\right| < \delta_2 \; \textit{for all } h' \varepsilon \eta_2 \quad \textit{and } 1 \leq i \leq 2d+1 \quad ,$$

where δ_2 is the minimum of the five quantities

$$1 - \delta_1$$

$$K_1 - \delta_1$$

$$\frac{1}{2K_2} - \delta_1$$

$$K_3 - \delta_1$$

$$\frac{1}{3}\min\left[z_{i+1}^{(1)} - z_i^{(1)} : 1 \leq i \leq 2d\right] \quad ,$$

and we let $\delta_3 = \delta_1 + \delta_2$. The constraints on δ_2 and $h^{(1)}$ guarantee that

$$(*1) \quad h' x_{i+1} - h' x_i > z_{i+1}^{(1)} - z_i^{(1)} - 2\delta_2$$

$$> 0 \; \textit{for all } h' \varepsilon \eta_2 \textit{ and } 1 \leq i \leq 2d \quad .$$

Thus, monotonicity of the $h'x_i$'s is maintained for all $h' \varepsilon \eta_2$.

Finally we choose a basis $h_1, h_2, ..., h_d$ from η_2. Letting $z_{j,i} = h_j x_i$, for $1 \leq j \leq d$, and $1 \leq i \leq 2d+1$, the final set of first layer wieghts (A, α) is given by

$$A = A_1$$

$$\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_d)^T$$

where

$$A_1 = \begin{bmatrix} a_1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & a_2 & 0 & 0 & \cdots & 0 \\ 0 & 0 & a_3 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & a_d \end{bmatrix}_{d\,X\,d}$$

$$H_1 = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_d \end{bmatrix}_{d\,X\,d} .$$

The a_i's and α_i's are given by

$$a_1 = \frac{2}{z_{1,d+1} - z_{1,1}}$$

$$\alpha_1 = \frac{z_{1,1} + z_{1,d+1}}{z_{1,1} - z_{1,d+1}}$$

$$\left.\begin{aligned} a_j &= \frac{2}{z_{j,j+d+1} - z_{j,1}} \\ \alpha_j &= \frac{z_{j,1} + z_{j,j+d+1}}{z_{j,1} - z_{j,j+d+1}} \end{aligned}\right\} for\ 2 \le j \le d \quad .$$

The outputs $u_{j,i}$, for $1 \le j \le d$ and $1 \le i \le 2d+1$, are defined by

$$u_{j,i} = p\left(a_j z_{j,i} + \alpha_j\right),$$

and take the following form

$$u_1 = (-1,-1,-1,\cdots,-1,-1)^T$$

$$u_2 = \left(-1+e_{1,2},-1+e_{2,2},-1+e_{3,2},\cdots,-1+e_{d-1,2},-1+e_{d,2}\right)^T$$

$$\vdots \qquad\qquad \vdots$$

$$u_d = \left(-1+e_{1,d},-1+e_{2,d},-1+e_{3,d},\cdots,-1+e_{d-1,d},-1+e_{d,d}\right)^T$$

$$u_{d+1} = \left(1,u_{2,d+1},u_{3,d+1},\cdots,u_{d-1,d+1},u_{d,d+1}\right)^T$$

$$u_{d+2} = \left(1,u_{2,d+2},u_{3,d+2},\cdots,u_{d-1,d+2},u_{d,d+2}\right)^T$$

$$u_{d+3} = \left(1,1,u_{3,d+3},\cdots,u_{d-1,d+3},u_{d,d+3}\right)^T$$

$$u_{d+4} = \left(1,1,1,\cdots,u_{d-1,d+4},u_{d,d+4}\right)^T$$

$$\vdots \qquad \vdots$$

$$u_{2d} = \left(1,1,1,\cdots,1,u_{d,2d}\right)^T$$

$$u_{2d+1} = (1,1,1,\cdots,1,1)^T \quad .$$

In order to prove the theorem it must be shown that

$$(*2) \quad \left\|p(Bu_i+\beta)-v_1\right\| < R_v \ for\ 1 \le i \le d \ ,$$

and

$$(*3) \quad p(Bu_i+\beta) = p\left(Bu_i^{(0)}+\beta\right) \text{ for } d+1 \le i \le 2d+1 .$$

The first inequality establishes the clustering of the points in S about v_1, while the second shows that the u_i's map into the same simplex V as do the $u_i^{(0)}$'s, $d+1 \leq i \leq 2d+1$. To this end we define the following quantities that bound the changes in the outputs between the mappings defined by ($A^{(0)}$, $\alpha^{(0)}$, and (A, α).

$$D_a = \max\left[\left|a_j - a_j^{(0)}\right| : 1 \leq j \leq d\right]$$

$$D_\alpha = \max\left[\left|\alpha_j - \alpha_j^{(0)}\right| : 1 \leq j \leq d\right]$$

$$D_u = \max\left[\left|u_{j,i} - u_{j,i}^{(0)}\right| : 1 \leq j \leq d, \text{ and } 1 \leq i \leq 2d+1\right]$$

$$D_u' = \max\left[\left|u_{j,i} - u_{j,i}\right| : 1 \leq j \leq d, \text{ and } 1 \leq i \leq d\right] \quad .$$

Invoking the upper bounds imposed on δ_1, and δ_2 the following inequalities can be proved.

(*4) $D_a \leq 2\delta_3 K_2^2$

(*5) $D_\alpha \leq 2\delta_3 K_2^2 z_{\max}^{(0)}$

(*6) $D_u \leq \delta_3 K_2\left(1 + 2\delta_3 K_2 + 4K_2 z_{\max}^{(0)}\right) \leq \dfrac{\delta_3 K_1}{K_3}$

(*7) $D_u' \leq 2\delta_3 K_2\left(1 + 2\delta_3 K_2\right) \leq 2D_u$

Proof of (*4).

For $1 \leq j \leq d$ we have

$$\left|a_j - a_j^{(0)}\right| = \left|\frac{2}{z_{j,j+d+1} - z_{j,1}} - \frac{2}{z_{j+d+1}^{(0)} - z_1^{(0)}}\right|$$

$$= 2\left|\frac{\left|\left(z_{j+d+1}^{(0)} - z_{j,j+d+1}\right) + \left(z_{j,1} - z_1^{(0)}\right)\right|}{\left(z_{j,j+d+1} - z_{j,1}\right)\left(z_{j+d+1}^{(0)} - z_1^{(0)}\right)}\right|$$

$$\leq 2\frac{2\delta_3}{\left(z_{j,j+d+1} - z_{j,1}\right)\left(z_{j+d+1}^{(0)} - z_1^{(0)}\right)}$$

$$\leq \frac{4\delta_3}{\left(z_{j,j+d+1} - z_1^{(0)} - 2\delta_3\right)\left(z_{j+d+1}^{(0)} - z_1^{(0)}\right)}$$

$$\leq \frac{4\delta_3}{\frac{1}{2}\left(z_{j+d+1}^{(0)} - z_1^{(0)}\right)\left(z_{j+d+1}^{(0)} - z_1^{(0)}\right)}$$

$$\leq \frac{8\delta_3}{\left(z_{j+d+1}^{(0)} - z_1^{(0)}\right)^2}$$

$$\leq \frac{8\delta_3}{\left(z_{d+1}^{(0)} - z_1^{(0)}\right)^2}$$

$$\leq \frac{8\delta_3.}{4 / K_2^2} = 2\delta_3 K_2^2 \quad .$$

Proof of (*5).

For $1 \le j \le d$ we have

$$\left|\alpha_j - \alpha_j^{(0)}\right| = \left|\frac{z_{j,1} + z_{j,j+d+1}}{z_{j,1} - z_{j,j+d+1}} - \frac{z_1^{(0)} + z_{j+d+1}^{(0)}}{z_1^{(0)} - z_{j+d+1}^{(0)}}\right|$$

$$= 2\frac{\left|z_{j,j+d+1} z_1^{(0)} - z_{j,1} z_{j+d+1}^{(0)}\right|}{\left(z_{j,1} - z_{j,j+d+1}\right)\left(z_1^{(0)} - z_{j+d+1}^{(0)}\right)}$$

$$= 2\frac{\left|z_1^{(0)}\left(z_{j,j+d+1} - z_{j+d+1}^{(0)}\right)\right| + z_{j+d+1}^{(0)}\left|\left(z_1^{(0)} - z_{j,1}\right)\right|}{\left(z_{j,1} - z_{j,j+d+1}\right)\left(z_1^{(0)} - z_{j+d+1}^{(0)}\right)}$$

$$\le 2\frac{2 z_{\max}^{(0)} \delta_3}{\left(z_{j+d+1}^{(0)} - z_1^{(0)} - 2\delta_3\right)\left(z_{j+d+1}^{(0)} - z_1^{(0)}\right)}$$

$$\le \frac{4 z_{\max}^{(0)} \delta_3}{\left(z_{d+1}^{(0)} - z_1^{(0)} - 2\delta_3\right)\left(z_{d+1}^{(0)} - z_1^{(0)}\right)}$$

$$\le \frac{8 z_{\max}^{(0)} \delta_3}{\left(z_{d+1}^{(0)} - z_1^{(0)}\right)^2} = 2\delta_3 K_2^2 z_{\max}^{(0)} \quad .$$

Proof of (*6).

For $1 \le j \le d$ and $1 \le i \le 2d+1$

$$\left|u_{j,i} - u_{j,i}^{(0)}\right| = \left|p\left(a_j z_{j,i} + \alpha_j\right) - p\left(a_j^{(0)} z_i^{(0)} + \alpha_j^{(0)}\right)\right|$$

$$\le \left|\left(a_j z_{j,i} - a_j^{(0)} z_i^{(0)}\right) + \left(\alpha_j - \alpha_j^{(0)}\right)\right|$$

$$\le D_\alpha + \left|a_j z_{j,i} - \alpha_j z_i^{(0)} + \alpha_j z_i^{(0)} - a_j^{(0)} z_i^{(0)}\right|$$

$$\le D_\alpha + \left|a_j\right|\left|z_{j,i} - z_i^{(0)}\right| + \left|z_i^{(0)}\right|\left|a_j - a_j^{(0)}\right|$$

$$\le D_\alpha + (K_2 + D_a)\delta_3 + z_{\max}^{(0)} D_a$$

$$\le 2\delta_3 K_2^2 z_{\max}^{(0)} + K_2\delta_3 + 2\delta_3^2 K_2^2 + 2 z_{\max}^{(0)} \delta_3 K_2^2$$

$$= \delta_3 K_2\left(1 + 2\delta_3 K_2 + 4K_2 z_{\max}^{(0)}\right) = \frac{\delta_3 K_1}{K_3} \quad .$$

Proof of (*7).

For $1 \le j \le d$ and $1 \le i \le d$

$$\left|u_{j,i} - u_{j,1}\right| = \left|p\left(a_j z_{j,i} + \alpha_j\right) - p\left(a_j z_{j,1} + \alpha_j\right)\right|$$

$$\le \left|a_j\left(z_{j,i} - z_{j,1}\right)\right|$$

$$\le \left|a_j\right|\left|\left(z_{j,i} - z_{j,1}\right)\right|$$

$$\le \left|a_j\right|\left|\left(z_{j,i} - z_i^{(0)}\right)\right| + \left|\left(z_i^{(0)} - z_1^{(0)}\right) + \left(z_1^{(0)} - z_{j,1}\right)\right|$$

$$\le \left|a_j\right|\left(\left|z_{j,i} - z_i^{(0)}\right| + \left|z_i^{(0)} - z_1^{(0)}\right| + \left|z_1^{(0)} - z_{j,1}\right|\right)$$

$$\le (K_2 + D_a)2\delta_3 = 2\delta_3\left(K_2 + 2\delta_3 K_2^2\right)$$

$$= 2\delta_3 K_2\left(1 + 2\delta_3 K_2\right) \le 2\delta_3 \frac{K_1}{K_3} \le 2D_u \quad .$$

Considering (*2) we have, for $1 \le i \le d$,

$$
\begin{aligned}
\left\| p(Bu_i+\beta) - v_1 \right\| &= \left\| p(Bu_i+\beta) - p(Bu_1+\beta) \right\| \\
&\le \left\| (Bu_i+\beta) - (Bu_1+\beta) \right\| \\
&= \left\| B(u_i - u_1) \right\| \\
&\le \|B\| \, \|u_i - u_1\| \\
&\le \|B\| \sqrt{d} D_u' \\
&\le \frac{r_v}{2K_1} 2D_u \\
&\le \left(\frac{r_v}{K_1} \right) \left(\frac{\delta_3 K_1}{K_3} \right) \\
&\le \frac{r_v \delta_3}{K_3} \le r_v \le R_v
\end{aligned}
$$

since $\delta_3 = \delta_1 + \delta_2 \le K_3$.

Proceeding with (*3) we have, for d+1 ≤ i ≤ 2d+1,

$$\left\| p(Bu_i + \beta) - p\left(Bu_i^{(0)} + \beta\right) \right\| \le \left\| B\left(u_i - u_i^{(0)}\right) \right\|$$

$$\le \|B\| \left\| u_i - u_i^{(0)} \right\|$$

$$\le \|B\| \sqrt{d} D_u$$

$$\le \left(\frac{r_v}{2K_1} \right) \left(\frac{\delta_3 K_1}{K_3} \right)$$

$$= \frac{r_v \delta_3}{2K_3} \le \frac{1}{2}$$

since $\delta_3 = \delta_1 + \delta_2 \le K_3$, and $r_v \le 1$.

Remark

No weight assignment for a (d,d,d)-PLN can effect the mapping guaranteed by this theorem when |Ext(X)| = d+1. In this case d of the d+1 members of T must be in Int(X).

INITIAL DISTRIBUTION

1 Commander in Chief, U. S. Pacific Fleet, Pearl Harbor (Code 325)
1 Commander, Third Fleet
1 Commander, Seventh Fleet
1 Naval War College, Newport
1 Headquarters, 497 IG/INT, Falls Church (OUWG Chairman)
2 Defense Technical Information Center, Alexandria
1 Center for Naval Analyses, Alexandria, VA (Technical Library)

ON-SITE DISTRIBUTION

1 Code C02
4 Code C0223 (3 plus Archives copy)
1 Code C023
20 Code C02307, Alltop
1 Code C02313, Chesnut
1 Code C27